

PATENT ABSTRACTS OF JAPAN

(11)Publication number : **2002-288110**

(43)Date of publication of application : **04.10.2002**

(51)Int.Cl.

G06F 13/14
G06F 9/445

(21)Application number : **2002-006721**

(71)Applicant : **MICROSOFT CORP**

(22)Date of filing : **15.01.2002**

(72)Inventor : **CHIOYAN JOHN H
MANN SAMUEL A**

(30)Priority

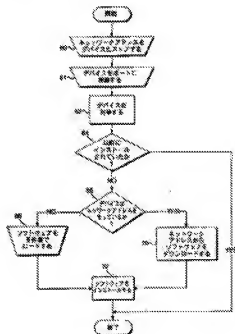
Priority number : **2001 760327** Priority date : **12.01.2001** Priority country : **US**

(54) METHOD AND SYSTEM FOR ACCESSING SOFTWARE RELATED WITH ELECTRONIC PERIPHERAL DEVICE BASED ON ADDRESS STORED IN PERIPHERAL DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To dispense with manually taking out information related with a peripheral device from CD-ROM.

SOLUTION: When the peripheral device 56 is connected to a host computer 20, a stored network address is read by a host computer 20. The network address is judged desirably by the host computer 20 when the peripheral device such as a USB device is connected to the host computer 20 at first or when power is supplied to the host computer 20 having newly connected USB. After this, the network address is used by the host computer 20 for communicating with a remote device, and it becomes possible to access information related with the peripheral device 56 such as a device driver for the peripheral device from a remote peripheral device 56 by the host computer 20.



(51) Int.Cl. ⁷	識別記号	F I	データコード* (参考)
G 0 6 F 13/14 9/445	3 3 0	G 0 6 F 13/14 9/06	3 3 0 B 5 B 0 1 4 6 1 0 C 5 B 0 7 6

審査請求 未請求 請求項の数37 O L (全 19 頁)

(21) 出願番号 特願2002-6721(P2002-6721)

(22) 出願日 平成14年1月15日 (2002.1.15)

(31) 優先権主張番号 09/760,327

(32) 優先日 平成13年1月12日 (2001.1.12)

(33) 優先権主張国 米国 (U S)

(71) 出願人 391055933
マイクロソフト コーポレーション
MICROSOFT CORPORATION
アメリカ合衆国 ワシントン州 98052-
6399 レッドモンド ワン マイクロソフト
ウェイ (番地なし)

(72) 発明者 ジョン エイチ. デロヤン
アメリカ合衆国 98052 ワシントン州
レッドモンド レアリー ウェイ ノース
イースト 15825 ユニット ビー105

(74) 代理人 100077481
弁理士 谷 義一 (外2名)

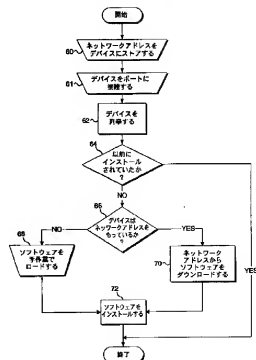
最終頁に続く

(54) 【発明の名称】 エレクトロニック周辺デバイスに関係するソフトウェアへのアクセスを周辺デバイスにストアされたアドレスに基づいて行う方法とシステム

(57) 【要約】

【課題】 周辺デバイスに関連する情報をCD-ROMから手作業で取り出さなくてもよいようにする。

【解決手段】 周辺デバイス56がホストコンピュータ20に接続されたとき、ストアされたネットワークアドレスはホストコンピュータ20によって読み取られる。ネットワークアドレスは、好ましくは、USBデバイスのような周辺デバイスが初めてホストコンピュータ20に接続されたとき、または新たに接続されたUSBをもつホストコンピュータ20に電源が入れたとき、ホストコンピュータ20によって判断される。そのあと、ネットワークアドレスは、リモートデバイスと通信するためにホストコンピュータ20によって使用され、周辺デバイス用のデバイスドライバといった、周辺デバイス56に関係する情報が、リモートの周辺デバイス56からホストコンピュータ20によってアクセスできるようにする。



【特許請求の範囲】

【請求項1】 周辺デバイスに關係する情報にアクセスする方法であって、

- (a) 周辺デバイスのストレージにネットワークアドレスを用意しておくステップと、
- (b) 周辺デバイスがホストデバイスに結合されているとき、ネットワークアドレスを周辺デバイスからホストデバイスの転送するステップと、
- (c) ホストデバイスと、ネットワークアドレスで示されたソースとの間の通信を可能にするステップであって、
10 そこでは前記通信は周辺デバイスに關係しているステップとを備えたことを特徴とする方法。

【請求項2】 請求項1に記載の方法において、前記用意しておくステップは、周辺デバイスのアドレス可能メモリにネットワークアドレスをストアしておくステップを含むことを特徴とする方法。

【請求項3】 請求項2に記載の方法において、前記転送するステップは、

- (a) 周辺デバイスのアドレス可能メモリ内にあって、ネットワークアドレスがそこにストアされたロケーションを指すポインタを用意するステップと、
- (b) ポインタをホストデバイスに伝達するステップと、
- (c) ポインタを使用して周辺デバイスのアドレス可能メモリ内のロケーションにアクセスし、ネットワークアドレスを前記ロケーションからホストデバイスに伝達するステップとを含むことを特徴とする方法。

【請求項4】 請求項1に記載の方法において、前記用意しておくステップは、周辺デバイスによって読み取り可能である取外し可能記憶媒体と再書き込み可能記憶媒体のうちの一方にネットワークアドレスをストアしておくステップを含むことを特徴とする方法。

【請求項5】 請求項1に記載の方法において、ホストデバイスに接続された周辺デバイスの数の変化を検出して、周辺デバイスがホストデバイスについて接続されたかを判断するステップをさらに含むことを特徴とする方法。

【請求項6】 請求項1に記載の方法において、周辺デバイスは、

- (a) ホストデバイス側の入出力ポートインタフェースとの有線のコネクションと、
- (b) ホストデバイスと周辺デバイスとの間のワイヤレスのコネクションのうちの一方を通してホストデバイスに結合されていることを特徴とする方法。

【請求項7】 請求項1に記載の方法において、前記通信を可能にするステップは、

- (a) ソースと通信するための許可をユーザに要求するステップと、
- (b) 前記要求をユーザから受け取ると、ホストデバイスとソースとの間の通信を開始して、周辺デバイスに關係する情報をソースから取得するステップとを含むこと
50

を特徴とする方法。

【請求項8】 請求項1に記載の方法において、前記転送するステップは、

- (a) ストリング記述子を得るための要求を周辺デバイスに対して発行するステップと、
- (b) ストリング記述子を受け取るステップと、
- (c) そのストリング記述子から、
10 (i) ネットワークアドレスと、
 (ii) ネットワークアドレスがストアされているロケーションを指すポインタのうちの一方を判断するステップと、を含むことを特徴とする方法。

【請求項9】 請求項1に記載の方法において、前記転送するステップは、

- (a) ストアされたネットワークアドレスを得るためにクラス要求を周辺デバイスに対して発行するステップと、
- (b) クラス要求に対する応答を受け取るステップと、
- (c) その応答から、
20 (i) ネットワークアドレスと、
 (ii) ネットワークアドレスがストアされているロケーションを指すポインタのうちの一方を判断するステップとを含むことを特徴とする方法。

【請求項10】 請求項1に記載の方法において、前記転送するステップは、

- (a) ネットワークアドレスを得るためにベンダ固有デバイス要求を周辺デバイスに対して発行するステップと、
- (b) ベンダ固有デバイス要求に対する応答を受け取るステップと、
- (c) その応答から、
30 (i) ネットワークアドレスと、
 (ii) ネットワークアドレスがストアされているロケーションを指すポインタのうちの一方を判断するステップとを含むことを特徴とする方法。

【請求項11】 請求項1に記載の方法において、前記通信を可能にするステップは、周辺デバイスに關係するデータ、マシン命令、およびドキュメントのうちの少なくとも1つを、ネットワークアドレスで示されたソースから自動的に取り出すステップを含むことを特徴とする方法。

【請求項12】 請求項1に記載の方法において、前記通信を可能にするステップは、ソースから取得され、周辺デバイスに關係するセットアッププログラムを自動的に実行するステップを含むことを特徴とする方法。

【請求項13】 請求項1に記載の方法において、前記通信を可能にするステップは、ネットワークアドレスで示されたソースに用意されているWebページを自動的に表示するステップを含むことを特徴とする方法。

【請求項14】 請求項1に記載の方法において、前記通信を可能にするステップは、周辺デバイスに關係するデバイスドライバプログラムをホストデバイスに自動的にインストールするステップを含むことを特徴とする方
50

法。

【請求項15】 請求項1に記載の方法において、前記通信を可能にするステップは、更新されたファームウェアを自動的にダウンロードして、周辺デバイスにインストールするステップを含むことを特徴とする方法。

【請求項16】 請求項1に記載の方法において、前記通信を可能にするステップは、周辺デバイスに関するアプリケーションプログラムを自動的に実行するステップを含むことを特徴とする方法。

【請求項17】 請求項1に記載の方法において、周辺デバイスのためのプロパティページであって、ネットワークアドレスへのリンクを含んでいるプロパティページを用意することにより、ユーザがそのリンクを選択してブラウザ機能をアクチベートし、ソースに順次アクセス可能とすることを特徴とする方法。

【請求項18】 請求項1に記載の方法において、前記伝達するステップは、ホストデバイス上でブラウザ機能を自動的に実行し、ネットワークアドレスへ自動的にナビゲートするステップを含むことを特徴とする方法。

【請求項19】 請求項1に記載の方法において、ユーザがホストデバイス上でブラウザ機能を実行することを望んでいるかどうかを要求し、そのユーザによって許可されているときだけ、ネットワークアドレスへ自動的にナビゲートするステップをさらに含むことを特徴とする方法。

【請求項20】 請求項19に記載の方法において、ホストデバイス上でブラウザ機能を自動的に実行し、ユーザによって許可されているならば、ネットワークアドレスへ自動的にナビゲートするステップをさらに含むことを特徴とする方法。

【請求項21】 請求項19に記載の方法において、ホストデバイス上でブラウザ機能を実行するさらに別の要求を、ユーザが選択的に禁止することを可能とし、それによりネットワークアドレスへ自動的にナビゲートするステップが実行されることを禁止するステップをさらに含むことを特徴とする方法。

【請求項22】 請求項1のステップ(b)と(c)を実行するマシン実行可能命令を格納していることを特徴とするマシン読取可能媒体。

【請求項23】 周辺デバイスに関する情報に自動的にアクセスするシステムであって、

- (a) ネットワークアドレスがストアされている周辺デバイスと、
- (b) (i) マシン命令がストアされているメモリと、
- (ii) ネットワーク上で通信するために使用されるネットワークインタフェースと、
- (iii) プロセッサと、
- (iv) その周辺インタフェースに動作可能に接続された周辺デバイスと通信するように構成された周辺インタフェースとを備えたホストデバイスと、

(c) 周辺デバイスに関係するマシン読取可能マテリアルのソースであって、該ソースは、周辺デバイスにストアされたネットワークアドレスに置かれたホストデバイスによって、ネットワークインタフェースを通して、アクセス可能であり、メモリにストアされた前記マシン命令は、

(i) 周辺デバイスがホストデバイスに結合されているとき、周辺デバイスからホストデバイスへネットワークアドレスを転送し、

(ii) ホストデバイスと、ネットワークアドレスに置かれたソースとの間の通信を可能にして、ホストデバイスがマシン読取可能マテリアルにアクセス可能にする処理をプロセッサに実行させるソースとを備えたことを特徴とするシステム。

【請求項24】 請求項22に記載のシステムにおいて、前記マシン命令により、さらに、

(a) スtring記述子を求める要求を周辺デバイスに対して発行し、

(b) String記述子を受け取り、

(c) そのString記述子から、

(i) ネットワークアドレスと、

(ii) ネットワークアドレスがストアされているロケーションを指すポインタのうちの1つを判断する処理を、プロセッサに実行させることを特徴とするシステム。

【請求項25】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、

(a) ストアされたネットワークアドレスを取得するためのクラス要求を周辺デバイスに対して発行し、

(b) クラス要求に対する応答を受け取り、

(c) その応答から、

(i) ネットワークアドレスと、

(ii) ネットワークアドレスがストアされているロケーションを指すポインタのうちの一方を判断する処理をプロセッサに実行させることを特徴とするシステム。

【請求項26】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、

(a) ネットワークアドレスを得るためにベンダ固有デバイス要求を周辺デバイスに対して発行し、

(b) ベンダ固有デバイス要求に対する応答を受け取り、

(c) その応答から、

(i) ネットワークアドレスと、

(ii) ネットワークアドレスがストアされているロケーションを指すポインタのうちの一方を判断する処理をプロセッサに実行させることを含むことを特徴とするシステム。

【請求項27】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、周辺デバイスに関するデータ、マシン命令、およびドキュメントのうちの少なくとも1つを、ネットワークアドレスで示されたソースからプロセッサに自動的に取り出させることを特

徴とするシステム。

【請求項28】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、ソースから取得され、周辺デバイスに関係するセットアッププログラムをプロセッサに自動的に実行させることを特徴とするシステム。

【請求項29】 請求項23に記載のシステムにおいて、前記マシン命令は、さらに、ネットワークアドレスで示されたソースに用意されているWebページをプロセッサに自動的に表示させることを含むことを特徴とするシステム。

【請求項30】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、周辺デバイスに関係するデバイスドライバプログラムをホストデバイスにプロセッサに自動的にインストールさせることを含むことを特徴とするシステム。

【請求項31】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、更新されたファームウェアをプロセッサに自動的にダウンロードさせ、周辺デバイスにインストールさせることを特徴とするシステム。

【請求項32】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、周辺デバイスに関係するアプリケーションプログラムをプロセッサに自動的に実行させることを特徴とするシステム。

【請求項33】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、周辺デバイスのプロパティページによって、ネットワークアドレスへのリンクを含んでいるプロパティページをプロセッサに用意させ、それにより、ユーザがそのリンクを選択してブラウザ機能を実行し、ソースに順次アクセス可能であることを特徴とするシステム。

【請求項34】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、ホストデバイス上でブラウザ機能を実行させる、ネットワークアドレスに置かれたソースに自動的にアクセスさせることを特徴とするシステム。

【請求項35】 請求項23に記載のシステムにおいて、前記マシン命令により、さらに、ユーザがホストデバイス上でブラウザ機能を実行することを望んでいるかどうかをプロセッサに要求させ、そのユーザによって許可されているときだけ、ネットワークアドレスに置かれたソースに自動的にアクセスさせることを特徴とするシステム。

【請求項36】 請求項35に記載のシステムにおいて、前記マシン命令により、さらに、ホストデバイス上でブラウザ機能を実行させる、ソースに自動的にアクセスさせる、ユーザによって以前に許可されている、ネットワークアドレスに置かれたソースに自動的にアクセスさせることを特徴とするシステム。

【請求項37】 請求項35に記載のシステムにおいて、前記マシン命令により、さらに、ホストデバイス上でブラウザ機能を実行するさらに別の要求を、ユーザが選択的に禁止することをプロセッサに可能にさせ、ネットワークアドレスに置かれたソースに自動的にアクセスするのを禁止させることを含むことを特徴とするシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般的には、エレクトロニック周辺デバイスのファームウェアメモリにネットワークアドレスをストアしておく方法とシステムに関し、さらに具体的に、かかるエレクトロニック周辺デバイスに具体的に関係するデバイスドライバや他のソフトウェアを、ネットワークアドレスによって特定されたりモットサイトから取り出すことに関する。

【0002】

【従来の技術】多くのコンピュータおよび関連の周辺デバイス(peripheral deviceまたはperipheral)は、コンピュータ上で稼動しているオペレーティングシステムに組み込まれたプラグアンドプレイ機能(plugin-and-play capability)を取り扱うように設計されている。プラグアンドプレイによると、コンピュータは新たに接続された周辺デバイスを自動的に検出し、その周辺デバイスをコンピュータシステムの一部としてインストールすることが可能になっている。プラグアンドプレイが開発される以前は、明示的に、新しい周辺デバイスが追加されたことを、ユーザがオペレーティングシステムに知らせ、どの周辺デバイスが追加されたかを特定し、例えば、割り

み要求値(interrupt request value - IRQ)のように、周辺デバイスが必要とするリソースを手作業でセットする必要があった。プラグアンドプレイの下では、コンピュータオペレーティングシステムは、新たに追加された周辺デバイスの存在を、コンピュータのブートアップ時に自動的に検出し、どのデバイスであるかを特定し、そのあと、そのデバイスが必要とするリソースを自動的に割り当てている。つい最近では、コンピュータがすでにブートアップされ、動作状態になった後で、周辺デバイスがいつ追加されたか、あるいは切り離されたかを「ホット検出(hot-detect)」し、どの周辺デバイスであるかを特定し、そのデバイスをコンピュータシステムの一部としてインストールし、あるいはアンインストールできるようにする機能が、オペレーティングシステムに追加されている。例えば、多くの新しいコンピュータと周辺デバイスはユニバーサルシリアルバス(Universal Serial Bus - USB)をサポートし、USBインタフェースを装備している。これらの新しいコンピュータ上で稼動しているMicrosoft(登録商標)社のWINDOWS(登録商標)オペレーションシステムの最新バージョンでは、スキャナ、プリンタ、デジタルカメラなどのような、新たに接続ま

たは切り離されたUSB周辺デバイスを、コンピュータの稼動中にホット検出することが可能になっている。オペレーティングシステムは、そのUSBデバイスがUSBポートに接続されたかを、そのデバイスから得た情報に基づいて特定している。そのあと、該当する周辺デバイスドライバとクライアントアプリケーションソフトウェアが以前にインストールされていれば、コンピュータは、そのデバイスをコンピュータシステムの一部としてインストールしている。具体的には、オペレーティングシステムは、その特定周辺デバイスドライバがカレントデバイスレジストリ(current device registry)に登録されているかどうかを確かめるためのチェックを行い、登録されていれば、対応するデバイスドライバをメモリにロードしている。そのあと、新たに接続されたUSB周辺デバイスは即時に使用可能になる。

【0003】さらに、新しいUSB周辺デバイスが初めてコンピュータに接続されるときは、そのUSB周辺デバイスはデバイスレジストリにはリストされていない。ある種の周辺デバイスでは、情報ファイル(つまり、*.INFファイル)とデバイスドライバはオペレーティングシステムに組み込まれているので、オペレーティングシステムは、周辺デバイスに関する必要情報をデバイスレジストリに追加し、デバイスドライバをロードすることが可能になっている。しかし、多くの場合、周辺デバイス情報とデバイスドライバはオペレーティングシステムに組み込まれていない。そのような場合には、現在では、周辺デバイスのための必要情報、ドライバ、および関連するクライアントアプリケーションソフトウェアを取得し、インストールするためには、セットアッププログラムを実行する必要がある。

【0004】どのタイプのアプリケーションソフトウェアをインストールする必要があるかの1つの例として、USBプリンタと共に使用されるプリンタマネージャプログラム(printer manager program)がある。そのほかにも、アプリケーションソフトウェアとしては、USBデバイスのための追加的フィーチャ(feature)や機能(function)を備えたプログラム、あるいは周辺デバイスの購入者に付加的便益を提供するプログラムがある。

【0005】代表例として、必要とされるドライバとユーザアプリケーションソフトウェアは、周辺デバイスと一緒にパッケージされているCD-ROMや1つまたは2つ以上のフロッピーディスクに入って提供されている。USB周辺デバイス用のドライバやユーザアプリケーションソフトウェアがオペレーティングシステムと一緒に提供されないときは、ユーザはCD-ROMまたはフロッピーディスクをロードするか、さもなければ、必要な材料を手作業で用意しなければならない。従って、周辺デバイスを初期インストールするためにやりとりすることがユーザに要求されるため、上級ユーザの経験が無駄になってしまうが、これは、周辺デバイスの初期インストールが真

の意味で「プラグアンドプレイ」オペレーションでないからである。

【0006】新しい周辺デバイスがコンピュータに接続されるとき、オペレーティングシステムは、プラグアンドプレイ機能を提供していれば、その新周辺デバイスの存在を検出する。上述したように、そのようなオペレーティングシステムは、そのあと、その周辺デバイスが以前にインストールされていたかどうかを確かめるためのチェックを行う。周辺デバイスが以前にインストールされていなければ、オペレーティングシステムは、新しい周辺デバイスをインストールするためのプロシージャを起動する。オペレーティングシステムが必要なファイルを備えていなければ、オペレーティングシステムは、該当する情報ファイル(例えば、*.INFファイル)、ドライバソフトウェア、およびアプリケーションソフトウェアが入っているCD-ROMまたは他の媒体を、ユーザに挿入させるためのオプションを用意している。

【0007】CD-ROMおよび/またはフロッピーディスクを作成し、それを周辺デバイスと一緒に配布することは、少なくとも2つの理由で、周辺デバイスベンダにとっては相当の費用負担になっている。第一の理由は、このような有形的コンポーネントを用意すると、媒体とパッケージングのコストが発生することである。第二の理由は、CD-ROMおよび/またはフロッピーディスクの作成をスケジューリングすると、新規に開発された周辺デバイスの出荷が遅れたり、利用できるソフトウェア開発時間量が少なくなったりすることである。開発サイクルの後半でデバイスドライバまたは他のソフトウェアの変更が必要になったときは、周辺ハードウェアの製造が完了した後までCD-ROMの作成が待たされることがある。周辺デバイスの出荷が遅延するほかに、完成した周辺デバイスを在庫として保管しておく費用が必要以上に増加することにもなる。これとは別に、周辺ハードウェアの製造と出荷スケジュールに合わせてCD-ROMを作成するためには、プログラム開発を途中で打ち切る必要が起るため、訂正を行ったり、プログラムフィーチャを追加したりすることができなくなる。プログラムフィーチャが除かれると、市場におけるプロダクトの競争力と魅力が低下することになる。また、ベンダは、更新されたソフトウェアを別のCD-ROMに入れて配布するか、あるいは別の方法で、バグを訂正したり、改善されたフィーチャと機能をソフトウェアマテリアルで提供したりすることが重要になる。

【0008】更新されたデバイスドライバと他のマテリアルは、インターネット(the Internet)上でダウンロードするために利用可能になっていることが多い。多くのコンピュータユーザは、高速で、フルタイムのコネクションでインターネットに結ばれているか、あるいは少なくともダイヤルアップ(dial-up)コネクションでインターネットに結ばれているので、周辺デバイスのベンダ

は、必ずしも、周辺デバイスと一緒にCD-ROMおよび/またはフロッピーディスクを提供しなくても、必要とされるドライバやアプリケーションソフトウェアが得られるようになっている。その代わりに、ペンドは、そのようなソフトウェアをインターネット上の特定サイトから手作業でダウンロードすることを、ユーザに要求できるようになっている。しかし、インターネットに接続し、および/またはドライバや他のソフトウェアマテリアルをダウンロードすることは、手作業による、時間のかかるプロセスでもあるため、大部分のユーザは避けようとしている。さらに、この手作業によるプロセスは、初心者ユーザにとっては問題となり、新規に購入された周辺デバイスをコンピュータに初期インストールすることに手間取ることによるため、周辺デバイスに対するユーザの満足度が低下することは明らかである。

【0009】

【発明が解決しようとする課題】以上の説明から理解されるように、好ましいことは、インターネット（または他のネットワーク）を通して、指定のリモートサイトから初めてコンピューティングデバイスに接続されたばかりの周辺デバイスに関連して、必要とされるデバイスドライバや関連のソフトウェア/ドキュメントマテリアルを、コンピューティングデバイス上のオペレーティングシステムが自動的に取得できるようにすることである。ユーザがCD-ROMやフロッピーディスクをホストコンピューティングデバイスに挿入したり、さもなければ、必要ソフトウェアを手作業で取得したりしないで済むようにすると、改善されたプラグアンドプレイ機能が得られることになる。ペンドがドライバや他のソフトウェアを提供するために、周辺デバイスと一緒にCD-ROMおよび/またはフロッピーディスクを提供しないでも済むようにすると、ソフトウェア開発時間が延長され、製造コストが低減されるはずである。

【0010】本発明は、周辺デバイスにストアされたネットワークアドレスを取得し、そのネットワークアドレスに置かれたリモートデバイスにアクセスして、周辺デバイスに関連する情報を得るようにした方法とシステムを目的とする。

【0011】

【課題を解決するための手段】周辺デバイスがホストコンピュータに接続されているとき、ストアされたネットワークアドレスはホストコンピュータによって読み取られ、ホストデバイスと、そのネットワークアドレスからアクセスされた周辺デバイスとの間の通信を可能にし、周辺デバイスに関連する情報がリモートデバイスからホストデバイスによってダウンロードされるようにしている。

【0012】ネットワークアドレスに置かれたリモートデバイスから情報をダウンロードすることのほかに、ホストデバイスは、リモートデバイスにストアされた、あ

るいはそこからダウンロードされたプログラムを自動的に実行し、周辺デバイス用のデバイスドライバをインストールし、アプリケーションプログラムをインストールし、Webページを表示し、周辺デバイスを登録し、周辺デバイスに関するドキュメンテーションをダウンロードし、ファームウェアをダウンロードして周辺デバイスにインストールし、ヘルプ情報にアクセスし、および/または周辺デバイスに関連する他のマテリアルにアクセスすることが可能になっている。当業者ならば理解されるように、周辺デバイスに関連するさまざまな、異種タイプの情報を、ネットワークアドレスからダウンロードし、表示し、あるいは取得することが可能である。

【0013】ネットワークアドレスは、好ましくは、周辺デバイスのリードオンリメモリ(read-only memory-ROM)にストアされる。しかし、別の実施形態では、ネットワークアドレスは、周辺デバイスに関連付けられていて、その周辺デバイスによって読み取り可能な再書き込み可能メモリ媒体(rewritable memory medium)にストアされている。

【0014】好ましくは、本発明の方法は、周辺デバイスがホストデバイスに接続されて、動作状態にあるとき周辺デバイスを検出して列挙(enumerate)するステップを含んでいる。この列挙を行うとき、周辺デバイスは自身であることを特定し、ホストデバイスへの通信の中でネットワークアドレスを転送する。通信は、ホストデバイス上のオペレーティングシステムが、周辺デバイスに割り当てられたデバイス記述子(device descriptor)を得るために周辺デバイスにクエリ（問い合わせ）したとき開始される。このデバイス記述子は、ペンド識別子(ID)とプロダクト識別子(ID)などの情報を含んでいる。また、オペレーティングシステムは、ストリング記述子(string descriptor)を得るために周辺デバイスにクエリする。周辺デバイスペンドは、ストリング記述子を周辺デバイスのメモリに埋め込む。そのストリング記述子には、周辺デバイスに関連するデバイスドライバ、ソフトウェア、ドキュメント、あるいは他のマテリアルをそこから取得できるネットワークアドレスが含まれている。別の方法として、ストリング記述子には、ネットワークアドレスが周辺デバイスにストアされているメモリロケーションを指すポインタを収めておくことも可能である。オペレーティングシステムは、ペンドID、プロダクトID、および他のIDを得るためにデバイス記述子を構文解析(parse)する。また、オペレーティングシステムは、ネットワークアドレスまたはそのネットワークアドレスを指すポインタを得るためにストリング記述子を構文解析する。

【0015】別の方法として、周辺デバイスからホストへネットワークアドレスを転送するステップは、ストリング記述子を周辺デバイスに要求して受け取る他の機能を利用することによって、列挙の後で実行することもで

11

きる。例えば、この転送ステップは、ネットワークアドレスを得るためのクラス要求を周辺デバイスに対して出すことによって実行することができ、この場合、クラスはシステム機能を動作させるために割り当てられたクラスであることが好ましい。さらに、別の方法として、この転送ステップは、周辺デバイスからネットワークアドレスを得るための1つまたは複数のベンダ固有コマンド(vendor-specific command)を実装することによって実行することもできる。

【0016】さらに、本発明の方法によれば、オペレーティングシステムによる予備的チェックも含まれ、周辺デバイスは以前にホストデバイスにインストールされていたかどうかを判断できるようにしている。そうでなければ、オペレーティングシステムは、デバイス記述子からネットワーク記述子を探し出し、ストリング要求を出し、ベンダ固有コマンドを出し、あるいは他の方法で、ネットワークアドレスを取得するようにしている。別の方法として、このチェックは、ホストデバイスがネットワークアドレスを取得した後で行うこともできる。この場合には、ネットワークアドレスにアクセスし、デバイスドライバおよび/または他の情報を取り出してインストールするタスクを起動するかどうか、および/またはネットワークアドレスにアクセスすることに関係する他のタスクを実行するかどうか判断される。

【0017】ホストデバイスとリモートデバイスとの間の通信を可能にするステップは、周辺デバイスに関係するデータ、マシン命令、またはドキュメントを、ネットワークアドレスに置かれたリモートデバイスを自動的に取り出すステップを含んでいる。好ましくは、通信を可能にするステップは、リモートデバイス側にストアされていて、周辺デバイスに関係するプログラムを実行する命令を自動的に受け取るステップを含んでいる。このステップには、周辺デバイス用のドライバをホストデバイスに自動的にインストールし、周辺デバイスに関係するアプリケーションプログラムを自動的に実行し、および/またはネットワークアドレスへのリンクを用意して、ユーザがリモートデバイス側に用意されている情報を選択できるようにするステップを、さらに含めることもできる。

【0018】本発明の方法によれば、好ましくは、ブラウザ機能(browser function)をホストデバイス側で自動的に実行し、ネットワークアドレスへ自動的にナビゲートするステップが、さらに含まれている。これとは別に、本発明の方法によれば、ユーザがブラウザ機能をホストデバイス側で実行することを望んでいるかどうかを要求し、その場合、ネットワークアドレスへ自動的にナビゲートするステップが含まれている。ユーザは、ブラ

12

ウザ機能をホストデバイス側で実行して、ネットワークアドレスへ自動的にナビゲートするための、さらに別の要求を禁止することが許されている。

【0019】本発明の別の態様によれば、周辺デバイスに関係する情報に自動的にアクセスするためのシステムを指向している。本発明のシステムは、マシン命令がストアされているメモリを装備したホストデバイスを含んでいる。ホストデバイス内のプロセスはマシン命令を実行し、これを受けて、ホストデバイスは、周辺デバイスにストアされたネットワークアドレスにアクセスする。ホストデバイスは周辺デバイスに接続されているので、リモートデバイスまたはマシン読取可能媒体の別ソースとの通信を可能にするネットワーク通信インタフェースも含んでいる。好ましくは、リモートデバイスは、ネットワークを利用してホストデバイスと通信するサーバになっている。しかし、リモートデバイスは、外部ストレージデバイスのような、別の周辺デバイスになることもできる。このようなリモートデバイスは、工業、自動車、または他のシステムで利用すると、情報を見つけるためにホストデバイスに使用させる識別子(ID)を用意しなくても、情報がどこにあるかを、周辺デバイスからホストデバイスに直接に知らせることができるので便利である。いずれの場合も、周辺デバイスに関係するデータ、マシン命令、ドキュメント、または他のマシン読取可能マテリアルは、リモートデバイスにストアされている。

【0020】本発明の別の態様は、上述した方法のステップを実行するマシン実行可能命令を格納としているマシン読取可能媒体を指向している。

【0021】以下では、本発明の上述した側面とそれに伴う利点の多くの理解を容易にするために、添付図面を参照して詳細に説明する。

【0022】

【発明の実施の形態】 例示の動作環境

図1と以下の説明は、本発明を実現することができる適当なコンピューティング環境の概要を、Webページをストアしてそれを提供するサーバと、そのWebページを要求してそれをユーザに表示するクライアントの両方と関連付けて要約して説明することを目的としている。必ずしもその必要はないが、本発明は、具体的には、プログラムモジュールのように、パーソナルコンピュータによって実行されるコンピュータ実行可能命令を中心に説明されている。一般的に、プログラムモジュールには、特定のタスクを実行し、あるいは特定の抽象データ型(abstract data type)を実装しているルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれている。さらに、当業者なら理解されるように、本発明は、特に、Webページを表示するクライアントデバイスに関しては、他のコンピュータシステム構成で実施することも可能であり、そのような構

13

成には、ハンドヘルドデバイス、ポケットパーソナルコンピュータ、ネットワークに接続するように構成されたデジタルセル電話、さらには、その他のマイクロプロセッサベースまたはプログラマブルコンピュータ、エレクトロニクスデバイス、ゲームコンソール、TVセットトップボックス、マルチプロセッサシステム、ネットワークパーソナルコンピュータ、ミニコンピュータ、メインフレームコンピュータ、工業制御機器、自動車機器、航空宇宙機器などが含まれている。また、本発明は、通信ネットワークを通してリンクされたリモート処理デバイスによってタスクが実行されるような、分散型コンピュータ環境で実施することも可能である。分散型コンピュータ環境では、プログラムモジュールは、ローカルとリモートの両方のメモリストレージデバイスに置いておくことができる。

【0023】図1を参照して説明すると、本発明を実現するための例示システムは、従来のパーソナルコンピュータ20の形態をした汎用コンピュータシステムを含む。処理ユニット21、システムメモリ22、およびシステムバス23を装備している。システムバスは、システムメモリを含む種々のシステムコンポーネントを処理ユニット21に結合しており、種々タイプのバス構造のいずれかにすることが可能であり、その中には、種々のバスアーキテクチャのいずれかを採用しているメモリバスまたはメモリコントローラ、周辺バス、およびローカルバスが含まれている。システムメモリには、リードオンリメモリ(read-only memory - ROM)24とランダムアクセスメモリ(random access memory - RAM)25が含まれている。基本入出力システム(basic input/output system - BIOS)26は、スタートアップ時のときのように、パーソナルコンピュータ20内のエレメント間の情報転送を支援する基本ルーチンから構成され、ROM24に格納されている。パーソナルコンピュータ20は、さらに、ハードディスク(図示せず)との間で読み書きを行うハードディスクドライブ27、取り外し可能磁気ディスク29との間で読み書きを行う磁気ディスクドライブ28、およびCD-ROMや他の光媒体などの、取り外し可能光ディスク31との間で読み書きを行う光ディスクドライブ30を装備している。ハードディスクドライブ27、磁気ディスクドライブ28、および光ディスクドライブ30は、それぞれ、ハードディスクドライブインタフェース32、磁気ディスクドライブインタフェース33、および光ディスクドライブインタフェース34を介してシステムバス23に接続されている。これらのドライブとそれぞれに関連するコンピュータ読取可能媒体は不揮発性ストレージとして、コンピュータ読取可能マシン命令、データ構造、プログラムモジュール、およびその他のデータをパーソナルコンピュータ20用に格納している。ここで説明している例示環境では、ハードディスク、取り外し可能磁気ディスク29、および取り外し

14

し可能光ディスク31が採用されているが、業者ならば理解されるように、コンピュータによってアクセス可能なデータを格納しておくことができる他のタイプのコンピュータ読取可能媒体を、例示の動作環境で使用することも可能であり、その中には、磁気カセット、フラッシュメモ리카ード、デジタルビデオディスク、ベルヌーイ(Bernoulli)カートリッジ、ランダムアクセスメモリ(RAM)、リードオンリメモリ(ROM)などが含まれている。

【0024】いくつかのプログラムモジュールは、ハードディスク、磁気ディスク29、光ディスク31、ROM24またはRAM25に格納しておくことが可能であり、その中には、オペレーティングシステム35(ブラウザ機能を含む)、1つまたは2つ以上のアプリケーションプログラム36(セットアッププログラムなど)、他のプログラムモジュール37、およびプログラムデータ38が含まれている。ユーザは、キーボード40やポインティングデバイス42などの入力デバイスを通して、コマンドや情報をパーソナルコンピュータに入力することができる。その他の入力デバイス(図示せず)として、マイクホーン、ジョイスティック、ゲームパッド、サテライトディッシュ、スキャナ、デジタルカメラなどがある。上記入力デバイスおよびその他の入力デバイスは、システムバスに結合された出力(I/O)デバイスインタフェース46を通して処理ユニット21に接続されていることが多い。プリンタ(図示せず)などの他の出力デバイスも、システムバスに結合されたI/Oデバイスインタフェース46を通して処理ユニット21に接続することが可能になっている。同様に、モニタ47や他のタイプのディスプレイデバイスも、ビデオアダプタ48などの、該当インタフェースを介してシステムバス23に接続され、Webページを表示するために使用可能になっている。モニタのほかに、パーソナルコンピュータは、スピーカなどの、他の周辺出力デバイス(図示せず)に接続されていることがよくある(図示していないが、サウンドカードや他のオーディオインタフェースを介して)。このような入出力デバイスは、周辺デバイス(peripheral device)と総称されている。本発明によれば、説明の便宜上、総称周辺デバイス56は、I/Oデバイスインタフェース46を介してパーソナルコンピュータ20に結合されている。I/Oデバイスインタフェース21の用語には、シリアルポート、パラレルポート、ゲームポート、キーボードポート、および/またはUSBポート用に使用される、各々の固有インタフェースが含まれている。I/Oデバイスインタフェース46を通して、情報は、システムバス23を経由して周辺デバイス56と処理ユニット21の間でやりとりされる。周辺デバイス56は、情報をストアしておくフームウェアメモリ58を備えている。フームウェアメモリ58としては、消去可能プログラマブルリードオンリメモリ(erasable programmable read-only memory - EPROM)、フラッシュ

15

メモリ、磁気ストレージ、その他のメモリがある。ファームウェアメモリ58には、周辺デバイス56に關係するベンダID、プロダクトID、シリアル番号、ファームウェア改訂(revision)レベル、および/または他の情報がストアされている。また、ファームウェアメモリ58には、周辺デバイスに關係するドライバ、データ、および/または他のマテリアルを、そこからアクセスし、および/またはダウンロードできるロケーションを示しているURIまたは他のタイプのアドレスもストアされている。

【0025】パーソナルコンピュータ20は、リモートコンピュータ49などの、1つまたは2つ以上のリモートソースとの論理コネクションを使用するネットワーク環境で動作させることが可能になっている。本発明によれば、周辺デバイスのメモリにストアされるURIまたはアドレスは、リモートコンピュータ49を通してアクセスされるロケーションを指定することができる。リモートコンピュータ49としては、別のパーソナルコンピュータ、サーバ(これは、全体的に、パーソナルコンピュータ20と非常によく似た構成になっているのが代表的である)、ルータ、ネットワークパーソナルコンピュータ、ピアデバイス、衛星、または他の共通ネットワークノードなどがあり、図1には、外部メモリストレージデバイス50だけが示されているが、パーソナルコンピュータ20に関連して上述したエレメントの多くまたは全部を装備しているのが代表的である。図1に示す論理コネクションには、ローカルエリアネットワーク(local area network - LAN)51と広域ネットワーク(wide area network - WAN)52が含まれている。このようなネットワーク環境は、オフィス、企業内(enterprise-wide)コンピュータネットワーク、イントラネット、およびインターネット(the Internet)で幅広く使用されている。

【0026】LANネットワーク環境で使用されるときは、パーソナルコンピュータ20は、ネットワークインタフェースまたはアダプタ53を通してLAN51に接続されている。WANネットワーク環境で使用されるときは、パーソナルコンピュータ20は、インターネットなどのWAN52上の通信を確立するためのモデム54または他の手段を装備しているのが代表的である。モデム54は内蔵型と外付け型があるが、どちらも、システムバス23に結合されているか、あるいはI/Oデバイスインタフェース46を介して、つまり、シリアルポートを通してバスに結合されている。ネットワーク環境では、パーソナルコンピュータ20に関連して示したプログラムモジュール、またはその一部は、リモートメモリストレージデバイスにストアしておくことができる。以上の説明から理解されるように、図示のネットワークコネクションは例示であり、コンピュータ間の通信リンクを確立するための他の手段を使用することも可能であ

16

り、そのようなものとしては、ワイヤレス通信(無線通信)とワイドバンドネットワークリンクがある。

【0027】

【実施例】本発明の実装例

以下では、ホストデバイスとしての働きをするパーソナルコンピュータ20と、プリンタ、モデム、デジタルカメラ、キーボード、および/またはコンピューティングデバイスと結合されて通信を行うように構成された他の、ほとんどのタイプのデバイス周辺デバイス56との間で通信を行うためにI/Oデバイスインタフェース46を使用した好適実施形態の実装例について説明する。本発明は、プラグアンドプレイの設計になっている周辺デバイスと関連して使用すると特に便利であり、さらに具体的に、USB規格(仕様)に準拠する任意の周辺デバイスのように、コンピュータにホット接続される設計になっている周辺デバイスで使用するに便利である。当業者ならば理解されるように、以下に示されている詳細は若干修正するだけで、プラグアンドプレイ以外のテクノロジーにも、RS-232Cシリアル、IEEE 1284パラレル、IrDA赤外線、その他の標準などの、周辺デバイスと他の通信形態にも適用することが可能である。このことから理解されるように、本発明は、ほとんどのタイプの通信プロトコルを使用してホストデバイスと通信する。ほとんどのタイプの周辺デバイスにも適用可能である。しかし、USBポートを通してコンピューティングデバイスに接続される設計になっている周辺デバイスは、以下の例では、本発明の説明と開示を単純化するために統一的に使用されている。

【0028】図2は、周辺デバイス用のデバイスドライバをインストールし、および/または周辺デバイスに關係するアプリケーションソフトウェアや他のマテリアルにアクセスする実施形態で採用されている全体ロジックを示す。本発明の重要なアプリケーションは、周辺デバイスが初めてコンピューティングデバイスに接続されるとき、リモートサイトに自動的にアクセスして、周辺デバイス用のデバイスドライバをダウンロードし、インストール可能にすることである。ステップ60では、ネットワークアドレスが周辺デバイスにストアされる。好ましくは、ネットワークアドレスは、デバイスの製造時か、あるいは少なくとも、周辺デバイスがエンドユーザに販売される前に、周辺デバイスのROMにストアされる。

【0029】本発明の別の実施形態は、既存の周辺デバイスに關係するものであり、周辺デバイスの識別子(ID)および/または他の情報は、フラッシュメモリ、電気的消去可能プログラマブルリードオンリメモリ(electrically erasable programmable read-only memory - EEPROM)、または他の形態の再書き込み可能不揮発性メモリ(この中には、フロッピーディスク、カートリッジ、メモリスティック、またはメモ리카ードのような、取り外し可能

17

および/または再書き込み可能記憶媒体も含まれることがある)内で更新または修正可能になっている。このようなデバイスでは、更新情報や代替情報は直接に得ることが可能である。取り外し可能または再書き込み可能記憶媒体を備えた周辺デバイスの例としては、ある種のプリンタ、ゲームコンソール、および工業用シンがある。望ましいことは、周辺デバイスに係るデバイスドライバおよび/または他のマテリアルを取り外し可能記憶媒体を使用しないで済むようにすることであるが、ネットワークアドレスを含んでいる新しい周辺デバイスで既存周辺デバイスを置き換えるのではなく、既存周辺デバイスをアップグレードした方が、経済的に有利であることがある。従って、場合によっては、取り外し可能および/または再書き込み可能記憶媒体からネットワークアドレスを既存周辺デバイスに追加することが可能である。この方法によると、ある種の既存周辺デバイスは、周辺デバイスにすでに入っている他の識別子(ID)の中から、ネットワークアドレスを含むようにアップグレードすることが可能になる。このように既存周辺デバイスを一度更新するだけで、その既存周辺デバイスはホストデバイスにネットワークアドレスを与えることが可能になり、この場合、ホストは、将来のデバイスドライバ修正版および/または他のマテリアルを得るために、引き続きいて別のCD-ROMや他の取り外し可能媒体に頼らなくても、将来のデバイスドライバ修正版と他の情報にアクセスすることが可能になる。このようにすると、既存周辺デバイスが、必要なデバイスドライバが存在しない別のホストデバイスに移された場合や、必要なデバイスドライバを取っているオリジナルCD-ROMが紛失した場合に特に好都合である。従って、ステップ60には、ネットワークアドレスを取っている取り外し可能または再書き込み可能媒体を周辺デバイスに装備させ、周辺デバイスに係る情報がそのネットワークアドレスからアクセス可能とすることが含まれている。

【0030】ステップ61では、ユーザは周辺デバイスをUSBポートに、つまり、I/Oデバイスインタフェース46(図1に図示)に接続する。別の方法として、ステップ61は、周辺デバイスがすでに接続されているホストコンピュータ20に電力を供給するようにすることもできる。さらに別の方法では、ステップ61を物理コネクションにするのではなく、論理コネクションにすることが可能である。このようなことは、周辺デバイスを、ホストコンピュータ20の赤外線線や無線周波数(RF)範囲内、あるいは他の通信範囲内にあるとされに行われている。

【0031】周辺デバイスがホストに接続されると、オペレーティングシステム35(図1に図示)は、ステップ62でその新デバイスを検出し、列挙する。前述したように、この列挙プロセス期間に、オペレーティングシステム35はデバイス記述子を得るために周辺デバイス

18

にクエリ(問い合わせ)する。デバイス記述子には、周辺デバイスに関する情報フィールドがあり、その中には、ベンダIDとプロダクトIDが含まれている。デバイス記述子のほか、コンフィギュレーション(構成)、インタフェースおよびエンドポイント記述子などの、他の標準記述子がデバイスに要求される。

【0032】同じく上述したように、オペレーティングシステム35は、ネットワークアドレス、またはそのアドレスを指すインデックスを得るために周辺デバイスにクエリする。ネットワークアドレスは、0xFFのように、周辺デバイス内の事前定義インデックスにストアしておくことができる。既知の事前定義または予約インデックスが使用される場合は、オペレーティングシステムは、ストリング記述子を求める標準要求でそのインデックスにクエリするだけで済むことになる。

【0033】別の方法として、オペレーティングシステムは、事前定義インデックスでクラス要求を出すことも可能である。クラス要求は、あるクラスの周辺デバイスによって理解されるコマンドである。例えば、プリンタはあるクラスの周辺デバイスを構成しているのに対し、モニタは別のクラスの周辺デバイスを構成している。準拠するプリンタは、標準要求のほかに、プリンタクラスに固有の要求を理解することができる。同様に、モニタは、標準要求のほかに、モニタクラスに固有の要求を理解する。しかし、URIは、周辺デバイスの種類に関係なく取得できるので、オペレーティングシステムクラスは、ネットワークアドレスを求める要求を追加するように構成されていることが好ましい。

【0034】記述子が分かると、オペレーティングシステム35は、その周辺デバイスは以前にインストールされていたかどうかを、判定ステップ64で判断する。例えば、オペレーティングシステムは、周辺デバイス情報がオペレーティングシステムのデバイスレジストリにリストされているかどうか、該当のデバイスドライバが使用可能であるかどうかをチェックする。このチェックには、周辺デバイスのファームウェア改訂番号がレジストリにリストされているものと同じであるかどうかを確認することを含めることができる。周辺デバイスが以前にパーソナルコンピュータ20にインストールされている(例えば、そのレジストリにリストされている)ならば、オペレーティングシステム35は、デバイスドライバおよび/または他のソフトウェアをメモリ22にロードするだけである。

【0035】周辺デバイスがパーソナルコンピュータ20上に登録されていないならば、あるいはデバイスドライバおよび/または他のソフトウェアがパーソナルコンピュータ20上で使用可能になっていないならば、オペレーティングシステム35は判定ステップ66を実行して、ネットワークアドレスが周辺デバイスにストアされているかどうかを判断する。この判断は、図3と図4を参照

19

して下述するように、いく通りかの方法で行うことができる。ネットワークアドレスが周辺デバイスにストアされていない場合は、デバイスドライバおよび/または他のソフトウェアは、従来技術で現在行われているように、ステップ68で、手作業でロードしなくてはならない。手作業によるローディングは、CD-ROMやフロッピーディスクのような記憶媒体を、該当ドライブに挿入して必要ソフトウェアをロードするか、あるいはインターネットを通してデバイスドライバおよび/または他のソフトウェアを手作業でダウンロードするか、あるいはユーザの介入を必要とする他のプロシージャを実行することによって行われるのが代表的である。手作業によるローディングは、ネットワーク通信がパーソナルコンピュータ20で利用できないか、あるいはアクティブ状態にない場合にも、必要になることがある。

【0036】他方、ネットワークアドレスが周辺デバイスにストアされていないば、オペレーティングシステム35はステップ70を実行して、ネットワークアドレスからアクセスされたリモートデバイスからデバイスドライバおよび/または他のソフトウェアのダウンロードを開始する。好ましくは、オペレーティングシステム35は、ネットワークアドレスを通してリモートデバイスとの通信を確立し、そのあと、以前に取得された周辺デバイス情報をリモートデバイスに渡すようにしている。例えば、オペレーティングシステム35は、プロダクトIDとファームウェア改訂レベルをリモートサーバに渡すことができるので、リモートサーバは、どのデバイスドライバ、アプリケーションソフトウェア、および/または他のマテリアルをダウンロードするのに相応しいかを、その情報を使用して判断する。デバイスに関する情報を周辺デバイスメーカーおよび/または企業のイントラネットサーバに渡すことを利用すると、重複するマテリアルを同じ企業に送信する必要性が少なくなり、さもなければ、同タイプの複数周辺デバイスのインストールを管理する必要性が少なくなることになる。しかし、好ましくは、マテリアルのダウンロード経路は、ハイパテキスト転送プロトコル(hypertext transfer protocol - HTTP)、ファイル転送プロトコル(file transfer protocol - FTP)、または他の通信方式で行われるファイル転送要求によって行われる。

【0037】最後に、ネットワークアドレスで参照されたサイトからダウンロードされたデバイスドライバおよび/または他のソフトウェアは、ステップ72でインストールされる。このインストールは完全に自動化することも、ユーザがインストールフィーチャをカスタマイズし、あるいは周辺デバイスまたは関連ソフトウェアのパラメータを選択できるようにする「ウィザード(wizard)」またはセットアッププログラムを使用して行うことも可能である。

【0038】次に、ネットワークアドレスが周辺デバ

20

スにストアされているかどうかを判断するステップが実装されている好適実施例について、図3を参照して説明する。上述したように、ステップ62で列挙を行うとき(図2)、オペレーティングシステム35(図1)は、いくつかの記述子を周辺デバイスに要求し、その中には、ユニフォームリソースID情報ストリング記述子が含まれている。上述したように、ストリング記述子は、ネットワークアドレス、そのネットワークアドレスを含んでいる別のユニフォームリソースID情報ストリングを指すポインタ、またはネットワークアドレスおよび/または追加URI情報を収めておくことができる。USBストリングは、オペレーティングシステム35が解釈できる、国際標準であるUNICODE(登録商標)でエンコード(符号化)されている。ユニフォームリソースID情報ストリング記述子および他の記述子は、RAM25(図1)にデータ80として格納されている。オペレーティングシステム35は、ステップ82で、別々のURIデータエレメントになるようにこれらの記述子を構文解析する。判定ステップ84では、オペレーティングシステム35は、値がゼロであるかどうかをチェックするか、あるいは周辺デバイスにネットワークアドレスが用意されていないことを示す他の事前定義値をチェックすることによって、ネットワークアドレスを指すポインタが存在するかどうかを判断する。構文解析した記述子からネットワークアドレスが判明しなかったときは、全体的判断の結果(図2のステップ66)がステップ86で“NO”にセットされたあと、インストールプロセスは続けられ、ドライバおよび/または他のソフトウェアがCD-ROMまたはフロッピーディスクから従来のように手作業でロードされるか、あるいはインターネットまたは他のネットワーク経由で、手作業でダウンロードすることによってロードされる(図2のステップ68)。しかるに、ネットワークアドレスが上記方法で判明したときは、全体的判断の結果(図2の判定ステップ66)はステップ88で、“YES”にセットされるか、あるいは正に符号化される。その結果、全体的インストールプロセスが続けられ、ネットワーク経由で自動的にロードされることになる(図2のステップ70)。

【0039】0xFFのように、URI情報ストリングで要求される事前定義または予約ストリング記述子インデックスを使用しないで済むようにするために、あるいはUSB標準記述子フォーマットを使用しなくても済むようにするために、図4に示す別の実施形態では、ネットワークアドレスが周辺デバイスにストアされているかどうか、そしてそれがどこにストアされているかを判断するようになっている。標準列挙プロセス期間に、ステップ90で、オペレーティングシステム35は、周辺デバイスのメモリに置かれているネットワークアドレスまたはネットワークアドレスを指すポインタを返すように、標準外ベンダ固有デバイス(Vendor Specific Device - VSD)要

求を周辺デバイスに対して発行する。VSD要求は、ベンダによって定義され、周辺デバイスによって認識されるコマンドである。USB標準では、ベンダが独自の要求コマンドを定義できるようにする要求コードが予約されている。具体的には、ユニバーサルリアルタイム規格（仕様）改訂版2.0、セクション9.3のテーブル9-2には、“bmRequestType” フィールドのビット5と6を2（バイナリ値では10）の値にセットすると、要求がベンダ要求であると特定されることが規定されている。代表例として、ベンダは周辺デバイスメーカであり、VSD要求は、製造時に使用されるように、あるいはインストール後は、周辺デバイスの通常オペレーション期間にデバイスドライバによって使用されるように、ベンダによって定義されている。このフィチャによると、周辺デバイスメーカは、周辺デバイスに固有のタスクを実行するように周辺デバイスに指示することができる。しかし、本実施形態では、ユニークなVSD要求は、ネットワークアドレスを得るためにオペレーティングシステム3.5によって出される。従って、どの周辺デバイスメーカも、オペレーティングシステム3.5に用意されているユニークなコマンド20を利用することが可能になっている。前記のUSB規格のセクション9.5によれば、コンフィギュレーション情報から独立しているか、あるいは標準外フォーマットを使用しているベンダ固有記述子が周辺デバイスから戻されるようになっている。

【0040】すでに述べたように、さらに別の実施形態では、オペレーティングシステムから出されたネットワークアドレス要求に対して、“bmRequestType”のビット5と6を1の値（バイナリ値では01）にセットすることによって新しいUSB「クラス」を設定するようにしている。どちらの場合も、VSDまたはクラス要求、例えば、GET_NETWORK_ADDRESSを用意することができ、列挙期間に、オペレーティングシステム3.5は、この要求を周辺デバイスに対して出すことが可能になっている。この要求を受けて、周辺デバイスはストアされたネットワークアドレスを戻すか、あるいは周辺デバイスのファームウェアメモリ内でネットワークアドレスがストアされているロケーションを指すポインタを戻している。周辺デバイスからネットワークアドレスまたはポインタがオペレーティングシステム3.5に戻されると、オペレーティングシステムはネットワークアドレス、またはポインタで指されているロケーション（そこにネットワークアドレスがストアされている）を、ステップ9.2で読み取ってメモリに入れる。判定ステップ9.4では、オペレーティングシステム3.5は、読み取った値がゼロに等しいかどうか、あるいはネットワークアドレスが用意されていないことを示す他の事前定義値に等しいかどうかを判断する。そうであれば、全体的判断の結果（図2のステップ66）は、ステップ86で“NO”にセットされる。そうでなければ、全体的判断の反対結果（ステップ66）

は、ステップ88で“YES”にセットされる。

【0041】本発明のもっと詳細な実施形態は図5に示されている。この実施形態では、本発明は、Microsoft社のWINDOWS（登録商標）オペレーティングシステムに組み込まれている。具体的には、USB周辺デバイスのホット接続インストールを受け持つ部分は、オペレーティングシステムに組み込まれている。上述したように、USB周辺デバイスのホット接続機能と後続の列挙は、ユーザがUSBデバイスを、パーソナルコンピュータ2.0などのホストコンピューティングデバイスに接続するか、あるいはUSB周辺デバイスが接続されたままでホストコンピューティングデバイスに電源が入られたとき、アクチベートされる。ステップ110と112は、これらの代替実施形態を示している。ステップ114では、バスドライバ(BusDriver)プログラムは新しいUSB周辺デバイスを検出し、バスに接続されたデバイスの数が変わったことをプラグアンドプレイマネージャ(Plug-and-Play Manager)プログラムに通知する。ステップ116では、プラグアンドプレイマネージャは、デバイス記述子と他のデバイス機能を得るために新周辺デバイスにクエリするI/O要求パケット(I/O request packet - IRQ)を送信することによってそのデバイスを列挙する。詳細な列挙タスクは、プラグアンドプレイマネージャプログラムと協力してバスドライバプログラムによって実行される。バスドライバプログラムは、PCI (peripheral component interconnect - 周辺コンポーネント相互接続) またはPCMCIA (Personal Computer Memory Card International Association - パーソナルコンピュータメモリーカード国際協議会)の親バスドライバであるのが代表的である。この列挙プロセスは、プラグアンドプレイマネージャが、そのベンダIDやプロダクトIDのように、周辺デバイスに関する情報を収集する点で、上述したものと類似している。同じく上述したように、この時点で、プラグアンドプレイマネージャは、もし存在すれば、ネットワークアドレスと他のストリング記述子を周辺デバイスから取得する。

【0042】この実施形態では、ネットワークアドレスは、インターネットリソースに関するWorld Wide Web Consortium標準に準拠するURIの形態になっている。ネットワークアドレスを特定するURIストリングのほかに、他のサポートURI情報を周辺デバイスから取得することが可能になっている。例えば、ユーザフレンドリな名前やタイトルを取得して表示することも、URIの使用に関してオペレーティングシステム3.5に指示するための別の指示を取得することも可能になっている。この追加情報が周辺デバイスにストリング記述子としてストアされていれば、これは事前定義ストリングインデックスとして得ることができる。別の方法として、この追加情報は、上述したように、ClassドライバまたはVSD要求を通して取得することが可能である。また、上述したよう

に、情報ストリングは、直接に取得することも、周辺デバイスのファームウェアメモリ内のロケーションを指すポインタを通して取得することもできる。1つまたは2つ以上のストリング間で別の符号化を行うと、種々タイプの情報を区別することができる。例えば、前向きスラッシュ記号(/)を使用すると、情報のタイプまたはその後続く文字を情報のタイプとして扱うコマンドを示すことができる。スラッシュ1(I)は、その後続く文字が表示されるタイトルまたはテキストであることを示すことができ、スラッシュ1(I)は、オペレーティングシステムに対する指示を示すことができ、スラッシュ5(S)は、URIストリングを示すことができる。従来と同じように、これらタイプのコマンドの1つのあとに続く文字は、引用符で囲まれた実際のストリングにすることも、周辺デバイスのファームウェアメモリ内でストリング文字がストアされている別のアドレスを指すポインタにすることも可能である。

【0043】周辺デバイスから周辺デバイスIDと他の追加情報を取得すると、プラグアンドプレイマネージャは、以前にインストールされた周辺デバイスのレジストリを、ステップ118で検査する。判定ステップ120では、プラグアンドプレイマネージャは、新たに接続された周辺デバイスが以前にインストールされていたかどうかを判断する。そうでなければ、プラグアンドプレイマネージャは、その新デバイスドライバをロードするオペレーションをステップ122で実行する。これとは逆に、周辺デバイスが以前にインストールされていた場合、プラグアンドプレイマネージャは、インストールされているデバイスドライバをステップ124でロードする。この場合も、プラグアンドプレイマネージャは、Device-URIダイナミックリンクライブラリ(dynamic link library - DLL)をステップ126で実行することによってURIを使用すれば、そのURIを使用するブラウザや他の機能を使用することができる。

【0044】図6は、図5のステップ122に示すように、新デバイスドライバをロードするときの詳細を示す図である。最初に、プラグアンドプレイマネージャは、ステップ130に示すように、周辺デバイスURIストリング情報も含めて、デバイスIDデータをデバイスのレジストリにセーブする。ステップ132では、プラグアンドプレイマネージャはNew-Device DLLを実行し、例えば、インターネット上のURIで特定されたリモートサイトからダウンロードするといった方法で、ネットワークコネクション上のURIからデバイスドライバが取得される。ステップ134では、New-Device DLLは、デバイスのレジストリからURIと他のデバイスIDおよび情報をロードする。このステップを実行するために、New-Device DLLは、セットアップアプリケーションプログラムインタフェース(application program interface - API)とコンフィギュレーションマネージャAPI機能をコールす

る。これらの機能は、新しく接続された周辺デバイス用の可能な限りのデバイスドライバのリストを作るために使用される。このリストは、ホストマシン側またはURLにストアされている情報(*.INF)から作られる。判定ステップ136では、New-Device DLLは、デバイスURIがレジストリに存在するかどうかを確認する。周辺デバイスにURIが用意されていないときは、ユーザは手作業でインストールを行わなければならない。URIがレジストリにすでに存在していれば、オプションとして、New-Device DLLは、ステップ138でそのURIをユーザに表示することができる。さらに、オプションとして、New-Device DLLは、そのURIをアクチベートするかどうかをユーザに選択させるか、あるいは手作業のインストールをそのまま続けさせることもできる。判定ステップ140でユーザがURIをアクチベートしたか、あるいはオペレーティングシステムがURIを自動的にアクチベートしたときは、New-Device DLLはステップ142でブラウザ機能を実行し、URIがブラウザのアドレスラインにロードされる。ブラウザ機能が得られたURI、例えば、http://www.microsoft.com/hardware/update/default.htmを使用しリモートサイトにアクセスすると、New-Device DLLはそこからデバイスドライバを取得することができる。

【0045】当業者ならば理解されるように、URIはアクティブサーバページ(active server page - ASP)にすることも、他のサーバ側機能にすることも可能である。また、上述したように、ファイル転送プロトコル(FTP)通信を開始するときにURIを使用し、リモートサイトからデバイスドライバを取得することもできる。従って、ステップ142では、デバイスドライバファイルをパーソナルコンピュータ20に転送することも、あるいは*.INFファイルのロケーションを示すだけにしてNew-Device DLLがデバイスドライバを見付けるようにすることも可能であり、これは、オペレーティングシステムから要求されたとき、ユーザがデバイスドライバの更新ロケーションを手作業で入力するときに行われるのと同じである。ステップ144では、New-Device DLLは、周辺デバイスのインストールを行い、ダウンロードまたは見付けられたデバイスドライバが選択可能デバイスドライバのリストに組み入れられる。最良デバイスドライバが選択され、パーソナルコンピュータ20にインストールされると、プラグアンドプレイマネージャは、ステップ146でそのデバイスドライバをメモリにロードするので、周辺デバイスの使用が可能になる。

【0046】図7に示すように、URIを使用すると、URIに置かれたアプリケーションプログラムをアクチベートし、URIに置かれたWebページを表示し、あるいはURIに用意されている追加情報やサービスを得ることもできる。そのために、プラグアンドプレイマネージャは、URLがレジスタに存在するかどうかを判定ステップ150

で確める。存在しなければ、追加情報やサービスがないので、得ることができない。URIが存在すれば、URIをアクチベートすることを求めるプロンプトがユーザに出されるのを禁止するように、フラグがセットされていたかどうかを判定ステップ152で判断される。このフィッチャは、例えば、広告がURIから表示されるのをユーザが望んでいない場合に使用すると、便利である。「再要求しない(don't ask again)」オプションが以前に選択され、ブラウザ機能のチェックボックスのエントリが、リモートサイトに置かれたWebページを表示していれば、URIをもつブラウザ機能をアクチベートすることを求めるプロンプトは、ユーザに出されないことになる。別の方法は、URIをもつブラウザ機能をアクチベートすることを求めるプロンプトをユーザに出すのを禁止し、URIをもつブラウザ機能を自動的にアクチベートするだけにすることである。しかし、好ましい方法は、URIをもつブラウザ機能をアクチベートするオプションをユーザに与えることである。

【0047】「再要求しない」フラグが偽であれば、つまり、チェックボックスがユーザによって以前に選択されているければ、プラグアンドプレイマネージャは、ステップ154でDevice-URI DLLを実行する。ステップ156では、Device-URI DLLは、周辺デバイスにURIがあることをユーザに通知し、そのURIをアクチベートしたいのかどうかを求めるプロンプトをユーザに出す。そのプロンプトでは、ユーザは、オプションとして、チェックボックスをチェックして、「再要求しない」フラグをセットすることもできる。ユーザがプロンプトに対する応答を送ると、ユーザがチェックボックスをチェックしたかどうか、判定ステップ158で判断される。そうであれば、Device-URI DLLは、ステップ160で「再要求しない」フラグを真にセットする。チェックしていなければ、Device-URI DLLは、ステップ162で「再要求しない」フラグを偽にセットする。判定ステップ164では、Device-URI DLLは、ユーザがURIをアクチベートすることを選択したかどうかを、さらに判断する。選択していなければ、プロセスは完了する。しかし、選択していれば、Device-URI DLLは、ステップ166でブラウザ機能をアクチベートし、URIで指定されたネットワークアドレスへナビゲートする。前述したように、URIで表されたサイトには、ブラウザ機能にデバイスドライバを表示させ、広告を表示させ、アプリケーションプログラムのダウンロードとセットアップを開始させ、ヘルプ情報を提供させ、周辺デバイスに関するユーザマニュアルをダウンロードさせ、あるいは他のサービスと情報を提供させるものもある。

【0048】図8は、デバイスURIを別の利用の仕方をするために採用されたステップを示す図である。このフィッチャによると、ユーザはクリックすると、いつでもURIをアクチベートできるような形でデバイスURIを表示

することができる。ステップ170では、ユーザは、Microsoft社のWINDOWS（登録商標）オペレーティングシステムのコントロールパネルから選択できるデバイスマネージャ(Device Manager)プログラムを実行する。このデバイスマネージャは、システムに接続されたすべてのハードウェアデバイスのリストを表示し、その中には、URIを取っている周辺デバイスが含まれている。ユーザは、ステップ172で周辺デバイスを表すアイコンをクリックし、その周辺デバイスのプロパティを表示することを選択する。ステップ174では、デバイスマネージャは、デバイスプロパティシートの表示を開始し、選択した周辺デバイスに関連する種々のプロパティページへは、タブで移るようになっている。判定ステップ176では、デバイスマネージャはデバイスのレジストリからデバイスURIストリングを探し出す。選択した周辺デバイスのレジストリにURIストリングがあれば、デバイスマネージャはURIプロパティページを呼び出し、そこには、URIストリングをクリックすると、URIで指定されたネットワークアドレスとの通信がアクチベートされるようなフォーマットで、デバイスURIストリング(1つまたは複数)が表示されている。好ましくは、ステップ178では、このオペレーションは、ブラウザ機能を実行し、URIで指定されたロケーションまたはWebページへナビゲートすることによって行われる。URIがアクチベートされると(またはURIストリングがレジストリになければ)、デバイスマネージャはステップ180を実行し、標準周辺プロパティページを作成する。

【0049】図9を参照して説明すると、図には、特殊目的コンピューティングデバイスに関する、本発明の別の応用が示されている。このコンピューティングデバイスはパーソナルビークルアシスタント(personal vehicle assistant - PVA)220になっている。PVAは、いくつかの点でパーソナルデータアシスタント(personal data assistant - PDA)に類似している。しかし、PVAは、自動車のような車両で使用するのに最適化されている。マイクロプロセッサは、最新車両のオペレーションでますます重要な役割を果たし、車両、その照明システム、エンターテインメントシステムのオペレーションを制御するために使用され、最近では、緊急アシスタンスとナビゲーション機能を提供するのに使用されている。ドライバのドライビング時間がより安全に、より生産的に、より楽しいものになるかは、車両に搭載されたこれらの高度なフィッチャにかかっている。現在市販されている車両および開発中の車両の設計には、ディスプレイスクリーン、スピーカ、セルラ電話に結合されたリモートマイクロホン、デジタルナビゲーションマップのドライバ、およびWebアクセスといったような、エレクトロニクスフィッチャが組み込まれている。車両では、各周辺デバイス用の個々のワイヤを配線する必要があるので、各周辺デバイスをPVAに接続する費用がかなりの部

分を占めている。この問題を解消するために、周辺バスが使用され、多数の周辺デバイスが一組のワイヤ上でPVAと通信すること（および相互に通信すること）を可能にしている。

【0050】当業者ならば理解されるように、図9に示す実施例は、本発明が、パーソナルコンピュータおよび従来からパーソナルコンピュータと関連付けられている周辺デバイス以外にも、多数のタイプのコンピュータデバイスと周辺デバイスに適用可能であることを示しているにすぎない。本発明は、工業用プログラマブルロジックコントローラ、ビルディング管理コンピュータ、フライトコントロールコンピュータ、および、ほとんどのタイプの周辺デバイスとやりとりする他のインテリジェントデバイスといった、他の分野における特殊目的のコンピュータデバイスにも応用可能であることは、もちろんである。

【0051】PVA 220は、処理ユニット221、システムメモリ222、システムバス223、ストレージデバイスインタフェース224、バスインタフェース2246、およびネットワークインタフェース2253を装備している。ビデオアダプタ、プリンタインタフェースなどの他のコンポーネントを、オプションとしてPVA 220に装備させることも可能であるが、これらは、図面を簡単にするため示されていない。システムバスは、システムメモリを含む種々のシステムコンポーネントを処理ユニット221に結合しており、従来タイプのバス構造とアーキテクチャの1つまたはいくつかで構成することが可能であり、その中には、メモリバスやメモリコントローラ、周辺バス、およびローカルバスが含まれている。

【0052】システムメモリとしては、ROM 224とRAM 225がある。BIOS 226は、PVA 220内のコンポーネント間で情報を転送するために使用される基本ルーチンで構成され、ROM 224に格納されている。PVA 220は、さらに、ストレージデバイス227とのインタフェースとなっており、永続的または取り外し可能媒体との間で読み書きを行うようにしている。ストレージデバイス227は、ストレージデバイスインタフェース223を介してシステムバス223に接続されている。ストレージデバイスとその関連システム読取可能媒体は、PVA 220用のマシン命令、データ構造、プログラムモジュール、および他のデータをストアしている。当業者ならば理解されるように、プロセッサがアクセスできるデータをストアするために、多種類のマシン読取可能媒体が使用可能であり、そのようなものとしては、フロッピーディスク、磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ベルヌーイ(Bernoulli)カートリッジ、RAM、ROMなどがある。いくつかのプログラムモジュールは、マシン読取可能媒体に格納しておくことが可能であり、その中には、オペレーティングシステム2235（ブラウザ機能を含む）、1つまたは2つ以上のアプ

リケーションプログラム（セットアッププログラムなど）、他のプログラムモジュール237、およびプログラムデータ238が含まれている。

【0053】PVA 220は、バスインタフェース246と周辺バス252を介して周辺デバイスと通信する。周辺バス252は、IEEE 1394バス、ISO 11898バス（例えば、Bosch社のController Area Network）、または他のタイプのバスにすることができ、当業者ならば理解されるように、本発明を採用できる他の分野には、他にも多数の周辺バスが存在している。例えば、工業製造分野では、EN 50170バス（例えば、Profibus）、スマート周辺デバイス用のISA SP50バス（例えば、Fieldbus Foundationバス）が使用され、航空宇宙分野では、ライン交換可能ユニット用のARINC（登録商標）バスが使用されている。応用分野によっては、システムバス223を拡張するだけで済む場合もある。

【0054】なお、従来技術のオペレーティングシステムでは、新たに接続された周辺デバイスで使用できるデバイスドライバを探し出すために、特定のディレクトリを指定することが普通になっている。これに対して、本発明では、オペレーティングシステムは、デバイスドライバが置かれているロケーションを周辺デバイスから取得することが要件になっている。言い換えれば、周辺デバイスに用意されているロケーションまたはアドレスから、周辺デバイスに関する情報、例えば、周辺デバイス用のデバイスドライバが見つかるようにしている。本発明の使い方を示している図9の実施形態では、これと同じ新規機能が実装されている。

【0055】図9に示す周辺バス252には、デジタルセル電話233とGPS (global positioning system - 全球測位システム) 234が接続されている。周辺バス252に接続される周辺デバイスの代表例として、他にも、コンパクトディスクプレイヤ、デジタルビデオディスクプレイヤなどのように、現在自動車に内蔵されている周辺デバイスがある。デバイスドライバや他のマテリアルが周辺バス252に接続されているとき、これらに直接アクセスすることを可能にする追加ソース249を設けることも可能である。周辺デバイスを新たに接続することは、初期製造時にも、サービスステーションでの保守時にも、車両が使用状態に置かれた後の他の時期にも行うことができる。

【0056】その概要を上述したように、周辺バス252に接続される各周辺デバイスは周辺ノード256であり、情報を収めておくファームウェアメモリ258を装備している。ファームウェアメモリ258としては、EPROM、フラッシュメモリ、磁気ストレージ、あるいは他のタイプのメモリがある。ファームウェアメモリ258は、周辺バス256に關係するベンダID、プロダクトID、および/または他の情報をストアしている。そのほかに、ファームウェアメモリにストアされるものとし

て、周辺ノード用のドライバおよび周辺ノードに関係する他のマテリアルを、そこからアクセスし、および/またはダウンロードできるロケーションを示しているURIまたは他のタイプのアドレスもある。

【0057】従来と同じように、周辺ノードのファームウェアメモリにストアされるURIまたはアドレスは、そこから追加ソース249にアクセスできるロケーションを指定することができる。追加ソース249としては、別のPVA、外部サーバ（これはパーソナルコンピュータ20と同等機能を備えているのが普通である）、ルータ、ネットワークパーソナルコンピュータ、ピア周辺デバイス、衛星、あるいは別のネットワークノードがある。

【0058】パーソナルコンピュータ20と同じように、PVA220は、ネットワークインタフェース253とネットワーク251を介した1つまたは2つ以上のデバイスとの論理コネクションを使用するネットワーク環境で動作する。ネットワーク251はモバイル通信用のワイヤレス（無線）ネットワークにすることも、製造サイト、サービスステーション、プリントステーション、または居住地での通信用のワイヤド（有線）ネットワークにすることも可能である。ワイヤレスであるか、ワイヤドであるかに関係なく、ネットワーク251は広域ネットワーク(wide area network - WAN)、ローカルエリアネットワーク(local area network - LAN)、または他のタイプの通信ネットワークにすることができる。このようなネットワーク環境は、デジタルセルラ電話、GPS(global positioning system - 全球測位システム)、緊急ロードサイドアシスタンスシステム(emergency roadside assistance system)、 SHIPPINGフリー通信、およびインターネット用として広く使用されている。ネットワーク251を利用すると、PVA220は、追加ソース249のような、リモートソースと通信することができる。以上から理解されるように、図示のネットワークコネクションは例示であり、従来のモデムのように、通信リンクを確立する他の手段を使用することが可能である。

【0059】追加ソース249から周辺ノード256に関係する情報にアクセスすることは、USB実施形態に関して上述したのとまったく同じ方法で行われる。しかし、列挙および周辺ノード256との通信は、介在する周辺バス252を通して行われるので、処理ユニット221と周辺デバイス256の間に共通通信層を追加するだけで済むことになる。

【0060】新しい周辺ノードが周辺バス252に接続されたとき、オペレーティングシステム253は、USB実施形態に関して概要を上述したように、その新ノードを列挙する。しかし、新しい周辺ノードが周辺バス252に接続されたときは、バスインタフェース246は、最初に、その新周辺ノード用の周辺バスアドレスを設定

しなければならない。USB実施形態では、新デバイスが接続されるUSBポートについては、アドレスが分かっている。これに対して、周辺バスがPVA上のポートに接続されているだけである。従って、新ノードの列挙を可能にするには、各周辺ノード用の介在バスアドレスがその前に設定されていないなければならない。上述したように、このステップは、採用されている特定周辺バスに適用可能な周知標準に準拠して行われる。

【0061】例えば、新しいノードがIEEE 1394周辺バスに接続されるときは、バスは自身を完全にリセットし、周辺ノードのすべてが相互間で通信し合せて、バスに接続された各周辺ノード用の新バスアドレスを判断している。バスアドレスは、ゼロからn-1までの桁数になっており、ここでnは周辺ノードの数である。そのあと、PVA220は、周辺バス上の各周辺ノードを列挙し、その中には、周辺ノード256が含まれている。各周辺ノードは、USBにおけるデバイス記述子に類似したバス情報ブロックを収めている。バス情報ブロックの3番目と4番目のクワッドレット(quadlet)はグローバルにユニークなデバイスIDを収めている。

【0062】バス情報ブロックの外側のテキスト記述子は、URIをストリングとしてストアしている、ファームウェアメモリ258内の別のロケーションに置かれているリーフ(leaf)をポインタすることもできる。このテキスト記述子は、列挙時にも、列挙後にも取得することができる。ネットワークが周辺ノード256から取得されると、周辺ノードに関するネットワークアドレスと情報にアクセスするときの残りの方法ステップは、上述したUSB実施形態で実装されているステップとほぼ類似している。

【0063】以上、本発明の好ましい実施形態とその変形実施形態に関して本発明を説明してきたが、当業者ならば理解されるように、請求項に記載されている本発明の範囲を逸脱しない限り、本発明は種々態様に変更することが可能である。従って、本発明の範囲は、いかなる態様においても、上述した説明によって限定されるものではなく、その全体は請求項の記載に従って判断されるものである。

【図面の簡単な説明】

【図1】本発明を実現するのに適しているパーソナルコンピュータ(personal computer- PC)の例を示す概略ブロック図である。

【図2】本発明の好適実施形態に実装されているロジックを示すフローチャートである。

【図3】周辺デバイスは、そのネットワークアドレスが標準外デバイス記述子のフィールドとしてストアされているかどうかを判断する実施形態のロジックを示すフローチャートである。

【図4】周辺デバイスは、そのネットワークアドレスインデックスがストアされているかどうかを、ベンダ固有

デバイス(Vendor-Specific Device - VSD)要求を出すことによって判断する実施形態のロジックを示すフローチャートである。

【図5】ユニフォームリソース識別子(uniform resource identifier - URI)を使用して周辺デバイスをインストールし、および/または機能をアクチベートするために採用されたロジックを示すフローチャートである。

【図6】周辺デバイスに用意されているURIを使用して取得された新デバイスドライバをロードするために採用されたロジックを示すフローチャートである。

【図7】周辺デバイスから取得され、周辺デバイスに關係するURIを使用してブラウザ機能を選択的にアクチベートするために採用されたロジックを示すフローチャートである。

トである。

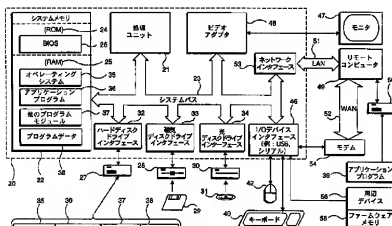
【図8】ユーザによって選択的に表示され、アクチベートされる周辺デバイスURIプロパティタグを作成するために採用されたロジックを示すフローチャートである。

【図9】本発明を実現するのに適しているパーソナルビークルアシスタント(personal vehicle assistant - PVA)システムの例を示す概略ブロック図である。

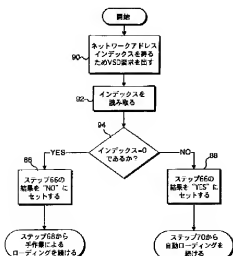
【符号の説明】

- 20 ホストコンピュータ
- 22 メモリ
- 25 オペレーティングシステム
- 46 I/Oデバイスインタフェース

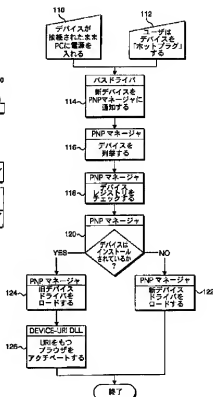
【図1】



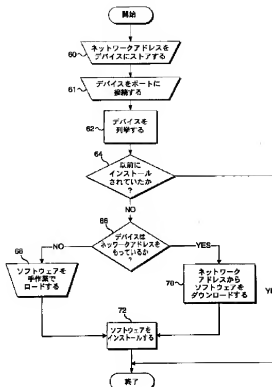
【図4】



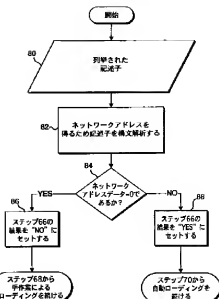
【図5】



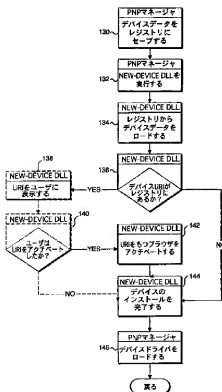
【図2】



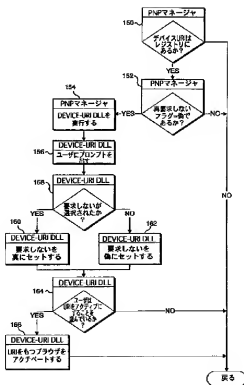
【図3】



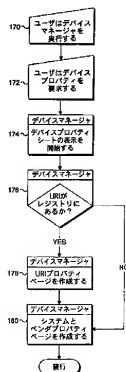
【図6】



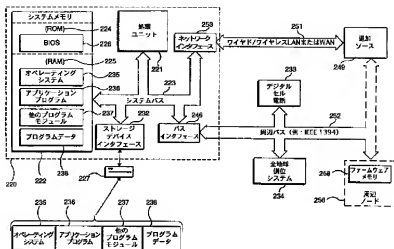
【図7】



【図8】



【図9】



フロントページの続き

(72) 発明者 サムエル エー. マン
 アメリカ合衆国 98008 ワシントン州
 ベルビュー レイク ヒルズ プールバー
 ド 222

Fターム(参考) 5B014 EB03 FB04 GD05 GD33 HC06
 5B076 AA06